Extended Abstract: Designer's Hardware Trojan Horse

Yousra Alkabani
CS Department
Rice University, Houston, TX
Email: yousra@rice.edu

Farinaz Koushanfar
ECE and CS Departments
Rice University, Houston, TX
Email: farinaz@rice.edu

***Abstract*—We introduce the first approach for pre-synthesis embedding of hardware Trojan horses (HTHs) by the designer. The embedded HTH gives the designer a low layer control over the hardware that can be exploited post-manufacturing. We identify the essential components for the exploit, outline the pre-synthesis implementation, and discuss detection. The new HTH embedding approach is low-overhead, effective, and hard to detect/remove. We emphasize that the diagnosis and isolation of designer's HTHs is a standing research challenge.**

I. Introduction

HTH is a component hidden inside a designed hardware which appears to perform a certain action but in fact performs a hidden task (at times). A number of instances of maliciously hidden embedded hardware and software by the designers has been reported. For example, in 2006, it was revealed that part of the software installed in the Vodafone Greece routers were altered by unknown entities. The modifications were such that the cell phone conversations of the prime minister and many other government officials were spied upon for more than two years, especially before and during the Olympic games in Greece [11]. As another example, the Seagate external hard drives recently contained pre-installed Trojans that transmitted the passwords back to a remote adversary [8]. Integrated circuits not only form the core for the industry, government and businesses, but also they are the underlying host for the programs and contents. Thus, the problem of spying Trojan circuits are of increasing importance.

There are many opportunities for Trojan horse insertion during the hardware design and implementation, system integration, system software, and programming. While software Trojans have been an active area of study, the work in hardware Trojans has been limited [2]. A number of emerging threats and possibilities for HTH insertion were discussed in the DoD's recent study on high performance microchip supply [1], including the Trojan insertion possibility by the foundry and embedding potential by the untrusted third-party and synthesis tools, but not by the designers. Recently, King et al. [9] proposed an approach to insert a designer's Trojan at the micro-architecture level. Even though such Trojans are not detectable above the OS level, they are detectable and removable at levels lower than the microarchitecture.

We introduce the designer's implanted hardware Trojans based on pre-synthesis manipulation of the circuit's structure by the designer, such that the hidden Trojan is blended into the post-synthesis design's structure. An abstracted view of the design process is presented in Figure 1, where the designer is shown on the left, and the design steps are organized from left to right. The designer composes the high level design description to find the computation model of the circuit that can be shown by a finite state machine (FSM). This is the phase where the designer has the most degree of freedom in modifying the circuit's functionality, without endangering a removal. The functional description of the circuit then goes through the standard flow until it is ready for fabrication.

The low-level modification gives the designer a unique edge for controlling the circuit. For example, the designer may devise HTH signals such that the authentication circuitry would be bypassed for certain hidden communications. Because of this low level control and also because of the fundamental lack of controllability and observability into the opaque internal of state-of-the-art integrated circuits, diagnosis and removal of HTHs are truly challenging. While verification tools are able to validate the states and transitions that are present in the public description, they are typically not able to check all the possible states. This is because the number of states grows exponentially with respect to the number of registers, typically in the order of 100s even for the small state-of-the-art ICs.

II. Foundation

We classify the components needed for a hardware Trojan horse exploit into three categories:

- *Trigger.* A trigger incites the planned Trojan horse action. This component could arrive from various internal (e.g., clock) or external (e.g., a sequence of inputs) sources.
- *Storage.* The action to be taken after a trigger occurs must be stored either explicitly in a memory component, or, implicitly by embedding it in another sequential circuit.
- *Driver.* A driver implements the action prompted by a trigger. Although the Trojan's act may be observable in some scenarios, in most others (e.g., spying) it will be hidden to normal users who interact with the system.

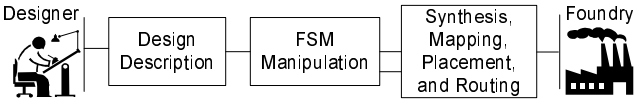

Fig. 1. The design flow of HTH.

The above components can be used in various ways to manage the implemented Trojans. Figure 2 presents an example flow for a typical hardware Trojan.

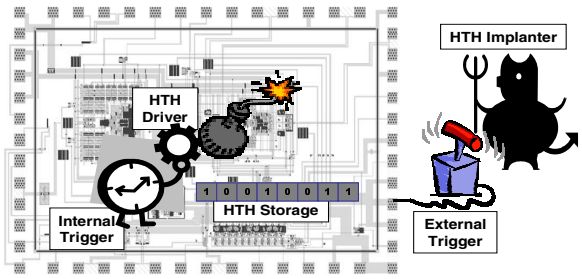

Fig. 2. The three components of HTH.

• **FSM manipulation.** Our Trojan horse embedding approach exploits the confidentiality of the design's functional abstraction known to the designer, for hiding purposes. Hiding at the functional level is done by altering the FSM and embedding states into it. By managing (controlling) the state-transitions inputs, one would be able to activate the hidden inputs in the design. The inputs may come from various triggers, that depend on the way the HTH is designed and the final goal of its implementation.

An important side benefit of exploiting the FSM for embedding additional functions is that it typically specifies the control part of the design, that is known to have less than 1% of the total area and power consumption of the whole circuit [7]. Therefore, even if the designer doubles or triples the FSM by embedding a large HTH in it, the overall overhead would still be low and insignificant for detection purposes. The low overhead of FSM manipulation was previously exploited for hiding watermarks within the design [10], to hide information in FSM [13], for active IC metering [3], [5], and for N-variant designs [4].

To embed the HTH in the design pre-synthesis, the designer first describes the Trojan components by using the FSM computational model. The FSM description should have a trigger as an input, and a driver hidden in the structure of the FSM. This FSM can be systematically hidden in the design by merging its states within the states of the original design's FSM. The basic idea of the merging is to ensure that both of the FSMs share the same FFs. Thus, the HTH would be inseparable (unremovable) from the original design's functionality.

• **Covert channels.** The hardware Trojan horse drivers must be implemented to be stealth to potential users. In many HTH applications where the goal is to extract confidential and user data from the working chips, the driver's result can be conveyed using the *covert channels* built into the output of the modified circuit. A covert channel is typically a stealth communication that uses the medium for legitimate communications. The common way to implement covert channels is to modify particular characteristics of the shared medium in a nonconventional and unforseen manner. Using the modification, the information is sent through the medium without being detected [6].

III. Detection

A key question that needs to be addressed is possibility of detection of the proposed implants. Diagnosis is truly challenging, if not impossible, from the perspective of the higher level users of the IC whose access can be bypassed by lower lever mechanisms. However, some implementation side-channels may help the high-level users in diagnosis. What exacerbates the detection problem is that the HTH functionality is conditioned upon the trigger's arrival. Thus, the HTH's functionality can be disabled during the testing phase for hiding purposes. Some invasive methods based on reverse-engineering the design's structure may have a better chance at detection. A possible way to address HTH diagnosis is to perform component-wise inspection of the design. For example, the communication channel and circuitry must be explicitly checked and verified for all states (if possible). Techniques for timing and storage covert channel detection may also be used to identify the spy's communication link [6], [12].

IV. Conclusion

We introduced the first systematic approach for embedding the designer's Trojan horse during the pre-synthesis phase of hardware designs. The Trojan embedding approach provided a low-level mechanism for bypassing the higher level authentication techniques. We discussed some possible side-channels that can help the diagnosis of the created hardware Trojans. We emphasize that detection and diagnosis of the introduced pre-synthesis Trojans is a standing challenge.

References

[1] Defense Science Board (DSB) study on high performance microchip supply, http://www.acq.osd.mil/dsb/reports/2005-02-hpms_report_final.pdf, 2006.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy*, pages 296–310, 2007.

[3] Y. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX Security*, pages 291–306, 2007.

[4] Y. Alkabani and F. Koushanfar. N-variant IC design: Methodology and applications. In *DAC*, 2008.

[5] Y. Alkabani, F. Koushanfar, and M. Potkonjak. Remote activation of ICs for piracy prevention and digital right management. In *ICCAD*, 2007.

[6] S. Cabuk, C. Brodley, and C. Shields. Ip covert timing channels: design and detection. In *ACM conference on Computer and communications security (CCS)*, pages 178–187, 2004.

[7] J. Hennessy and D. Patterson. *Computer architecture: a quantitative approach*. Morgan Kaufmann Publishers, 1996.

[8] G. Keizer. "maxtor drives contain password-stealing trojans", http://www.computerworld.com/action/article.do?command=viewarticlebasic&articleid=9046424, November 2007.

[9] S. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, and Y. Zhou. Designing and implementing malicious hardware. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[10] A. Oliveira. Techniques for the creation of digital watermarks in sequential circuit designs. *IEEE Trans. on CAD*, 20(9):1101–1117, 2001.

[11] V. Prevelakis and D. Spinellis. "the athens affair", http://www.spectrum.ieee.org/jul07/5280, July 2007.

[12] C. Tsai, V. Gligor, and C. Shandersekaran. On the identification of covert storage channels in secure systems. *IEEE Trans. Software Engineering*, 16(6):569–580, 1990.

[13] L. Yuan and G. Qu. Information hiding in finite state machine. In *Information Hiding*, pages 340–354, 2004.